



UPC Applications Overview

Parry Husbands
17 May 2004



Overview



- Our mission
 - Help improve Berkeley UPC Environment
 - Stress translator and runtime
 - Determine performance bottlenecks (if any)
 - Outreach
 - Showcase features of UPC
 - Continue explorations in PGAS languages
 - Validate design decisions
 - Question others
 - Suggest extensions



Our environment at scale



- NAS Multigrid Class D (1024 x 1024 x 1024)
- Written using 1-sided style

Threads	Mflop/sec
128	35,510
256	63,552
512	116,211
1024	202,141

Lemieux @ PSC
Alpha/Quadrics



Sparse Matrix – Vector Multiplication



- Compute $y = A \cdot x$ for distributed sparse matrix A
- Key component of solvers
- Irregular – threads require variable number of elements of x

<table border="1"><tr><td>X</td><td>X</td><td>X</td><td></td><td></td><td></td><td>X</td></tr><tr><td></td><td>X</td><td>X</td><td>X</td><td></td><td>X</td><td></td></tr></table>	X	X	X				X		X	X	X		X		*	<table border="1"><tr><td></td></tr></table>		=	<table border="1"><tr><td></td></tr></table>	
X	X	X				X														
	X	X	X		X															
<table border="1"><tr><td></td><td></td><td>X</td><td></td><td></td><td>X</td><td></td></tr><tr><td>X</td><td>X</td><td>X</td><td></td><td>X</td><td></td><td>X</td></tr></table>			X			X		X	X	X		X		X		<table border="1"><tr><td></td></tr></table>			<table border="1"><tr><td></td></tr></table>	
		X			X															
X	X	X		X		X														



SPMV using memcpy extensions



- Implemented using non-blocking indexed gets

`spmv(A,x)`

`// Compute Ax for sparse A`

`non-blocking get of remote values of x`

`computation of part of result using local values of x`

`sync`

`computation of rest of result using remote values of x`

- Significantly simplified code, particularly setup
- Not tuned yet, but shows promise



Outreach: Parallel Triangulation



- First step in many physical simulations
- Lots of dynamic data movement (~ 4K LOC + Triangle)



Improving the Delaunay Triangulation Code



- Message coalescing performed manually
 - Eventually automatic
- Teams library further optimized and extended
 - Now includes a distributed vector facility
- Fast redistribution code implemented

Time for 1 million points
in a sphere
HP UPC
Alpha/Quadrics@MTU

Threads	Time (s)
1	14.6
2	12.61
4	7.50
8	5.19



The effort continues...



- Message coalescing in practice
- Further improvements to Triangulation code
- More examples using the memcpy extensions
 - Performance tuning
- Programs using collectives
 - Are they the “right” interface?
- More involvement in language specification